

УДК 51-74

Н. П. Цыганенко

Белорусский государственный технологический университет

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМ КЭШИРОВАНИЯ

Работа посвящена математическому моделированию систем кэширования данных в веб-приложениях. Рассмотрена текущая ситуация на рынке веб-приложений относительно использования систем кэширования данных. Приведены основные варианты хранилищ, которые могут применяться при построении систем кэширования. Указаны их преимущества и недостатки. Также в статье вводится математический аппарат для описания таких систем. Выявлены основные характеристики систем кэширования. Рассмотрена проблема вычисления вероятности попадания в кэш, а также один из способов ее определения с использованием деревьев решений. Подробно освещены особенности процесса построения дерева решений для данного типа задач. Вместе с этим рассмотрены частные случаи, в которых вычисление данной вероятности тривиально. Приведена полученная в процессе научного исследования рекуррентная формула для нахождения вероятности попадания записей из базы данных в кэш. С использованием полученной рекуррентной формулы выведена формула для вычисления среднего времени чтения записи из системы кэширования. На основании полученной математической модели системы кэширования выведена целевая функция и описаны условия задачи оптимизации. Сделаны выводы о возможностях практического внедрения полученной математической модели и применения к ней методов оптимизации.

Ключевые слова: кэширование, математическая модель, оптимизация, дерево решений, веб-приложения.

N. P. Tsyganenko

Belarusian State Technological University

MATHEMATICAL MODELING OF CACHING SYSTEMS

This paper is devoted to mathematical modeling of data caching systems in web applications. We consider the current situation in the market of web-based applications for the use of data caching systems. The article also introduces the mathematical formalism to describe such systems. The main characteristics of caching systems are introduced. The problem of calculating the probability of falling into the cache, as well as one of the ways to calculate it using decision trees are described. The features of the process of building a decision tree for this type of tasks are studied in detail. At the same time we consider the special case in which the calculation of the probability is trivial. The recurrence formula produced in the process of scientific research is to calculate the probability that the records from the database cache. Using derived recurrence formula derived formula to calculate an average reading entries from the cache system time. Based on the caching system we produced a mathematical model of a target function and are described in terms of the optimization problem. Conclusion about the possibilities of practical application of a mathematical model and apply it to the optimization methods is made.

Key words: caching, mathematical model, optimization, decision tree, web applications.

Введение. Основная цель веб-приложений и других интернет-ресурсов заключается в предоставлении посетителям информации в различных формах (текст, графика, видео и т. д.). Интернет-пространство сегодня — это насыщенная данными среда. Поэтому, в условиях жесткой конкуренции за внимание пользователей, проблема скорости получения пользователем информации более чем актуальна. Ее решением является минимизация времени между запросом, посылаемым на сервер, и моментом полной загрузки ответного сообщения с запрашиваемым контентом.

Одной из точек минимизации времени получения информации является канал между серверным веб-приложением и реляционной базой данных. Для ускорения данного канала

используют механизм кэширования. Суть кэширования заключается в том, что часть информации из реляционной базы данных дублируется в хранилище с более высокой скоростью доступа — кэше. В момент, когда пользователь запрашивает информацию у веб-приложения, сначала проверяется кэш [1].

В случае если в нем необходимых данных не оказалось, запрос перенаправляется к реляционной базе данных. Таким образом, в случае если нужная информация находится в кэше, то получается значительный выигрыш во времени по сравнению с простым использованием базы данных.

Двумя наиболее важными характеристиками эффективности системы кэширования являются скорости доступа к хранимой информации и вероятность попадания в кэш. Скорость доступа к

информации в кэше – это время от запроса информации веб-приложением у системы кэширования до полного получения необходимых данных. Вероятность попадания в кэш – это процент успешных разрешений запросов на получение определенной информации к их общему числу.

Поскольку обе основные характеристики являются числовыми, то для того, чтобы получить наилучшие результаты при использовании систем кэширования, можно применить методы решения задач оптимизации. Эти методы относятся к математическому программированию и требуют наличия математической модели оптимизируемой системы в виде целевой функции [2].

Основная часть. Исходными данными при моделировании систем кэширования выступают множество записей в базе данных R , множество записей в кэше C и число обращений к базе данных U .

Множество записей в базе данных R мощностью n состоит из записей r_i :

$$r_i \in R, \quad (1)$$

где i – порядковый номер записи в базе данных.

Множество записей в кэше C мощностью m состоит из записей c_j :

$$c_j \in C, \quad (2)$$

где j – порядковый номер записи в кэше.

Множество записей в кэше C является подмножеством множества записей в базе данных R :

$$C \subset R. \quad (3)$$

Количество записей в кэше не может превышать количества записей в базе данных, и в большинстве веб-приложений объем базы данных в разы превышает объем кэша. Из этого следует, что мощность множества записей в кэше m не превышает мощности множества записей в базе данных n . Отсутствие системы кэширования может быть выражено пустым множеством с нулевой мощностью. Резюмируя вышеперечисленное, запишем ограничения для мощности множества записей в кэше:

$$0 \leq m \leq n. \quad (4)$$

Число обращений к множеству записей в базе данных состоит из количества обращений к отдельным записям:

$$U = \sum_{i=1}^n u_i, \quad (5)$$

где u_i – количество обращений к записи r_i .

Воспользовавшись формулой (5), выведем формулу для вычисления вероятности обращения к записи r_i через отношение количества

обращений к данной записи к общему числу обращений к базе данных:

$$p_i = \frac{u_i}{U}. \quad (6)$$

Поскольку порядковый номер записи в базе данных i представляет собой дискретную величину, то вероятность обращения к записям в базе может быть представлена в виде вектор-строки:

$$p = (p_1 \ p_2 \ \dots \ p_n). \quad (7)$$

Обозначим через P_i вероятность того, что запись r_i находится в кэше, т. е.

$$r_i \in C. \quad (8)$$

Если размер кэша равен размеру базы данных, то в кэш могут быть записаны сразу все записи из базы данных. Таким образом, вероятность попадания r_i в кэш равна 1.

Если размер кэша равен 0, т. е. система кэширования отсутствует, то в кэш не может быть помещена ни одна запись из базы данных. Итак, вероятность попадания r_i в кэш равна 0.

В случае когда p_i для всех i совпадает, т. е. обращение к любой записи равновероятно, вероятность попадания в кэш может быть вычислена следующим образом:

$$P_i = \frac{m}{n}. \quad (9)$$

В реальных веб-приложениях равномерного распределения в запросах к базе данных не наблюдается. Одни данные всегда запрашиваются чаще других. Это явление связано с наличием в базе данных служебной информации, а также с большей актуальностью и релевантностью определенных наборов данных, которые специфичны для каждого приложения.

Для построения математической модели систем кэширования необходимо вывести общую формулу для вычисления вероятности попадания r_i в кэш.

Рассмотрим частный случай. В базе данных имеется четыре записи: A, B, C, D. Вероятность того, что запросят конкретную запись, равна 0,1; 0,2; 0,3 и 0,4 соответственно. Размер кэша равен трем. Необходимо вычислить вероятность нахождения записи A в кэше.

Данная задача строится на вычислении условных вероятностей. Одним из возможных решений такого типа задач является построение дерева решений. Отличительной чертой данной задачи от типовых задач на нахождение условной вероятности является то, что события для которых вычисляется условная вероятность, находятся в одной группе.

На первом уровне дерева необходимо расположить узлы, соответствующие записям из базы данных, и соединить их с корнем. На ребрах следует записать вероятности того, что первой записью в кэше c_1 будет запись, к которой ведет ребро.

Далее для каждого из узлов первого уровня, кроме узла, обозначающего искомую запись, необходимо построить узлы второго уровня.

Для построения следующего уровня дерева решений для узла уровня i из исходного множества записей в базе данных извлекаются записи из узлов, которые ведут от узла i к корню, включая рассматриваемый узел. Полученное подмножество отражается в виде узлов, соединенных ребрами с рассматриваемым узлом. На ребрах записываются вероятности того, что $i + 1$ записью в кэше c_{i+1} будет запись, соответствующая узлу, с учетом того, что предыдущие записи в кэше уже известны.

Выведем формулу для расчета условных вероятностей, описанных выше. Пользуясь формулой (6), найдем выражение для вычисления количества обращений к записи r_i :

$$u_i = p_i U. \quad (10)$$

Вероятность того, что второй записью в кэше будет r_i при условии, что первой записью является r_j , может быть вычислена как отношение числа обращений к записи r_i к разности общего числа обращений и числа обращений к записи r_j :

$$p(i | j) = \frac{u_i}{U - u_j}. \quad (11)$$

Вероятность того, что третьей записью в кэше будет r_i при условии, что первой и второй записями являются r_j и r_k соответственно, может быть вычислена как отношение числа обращений к записи r_i к разности общего числа обращений и числа обращений к r_j и r_k :

$$p(i | j, k) = \frac{u_i}{U - u_j - u_k}. \quad (12)$$

Вероятности для остальных уровней дерева решений вычисляются по аналогии. Выражение (11) может быть записано через вероятность обращения к записи:

$$p(i | j) = \frac{p_i U}{U - p_j U}. \quad (13)$$

Если упростить, то получаем следующее соотношение:

$$p(i | j) = \frac{p_i}{1 - p_j}. \quad (14)$$

Дерево строится, пока число уровней не достигнет размера кэша. Корень является нулевым уровнем и используется для визуализации начального состояния, когда не просмотрена ни одна запись в кэше.

Дерево решений для описанной задачи представлено на рисунке.

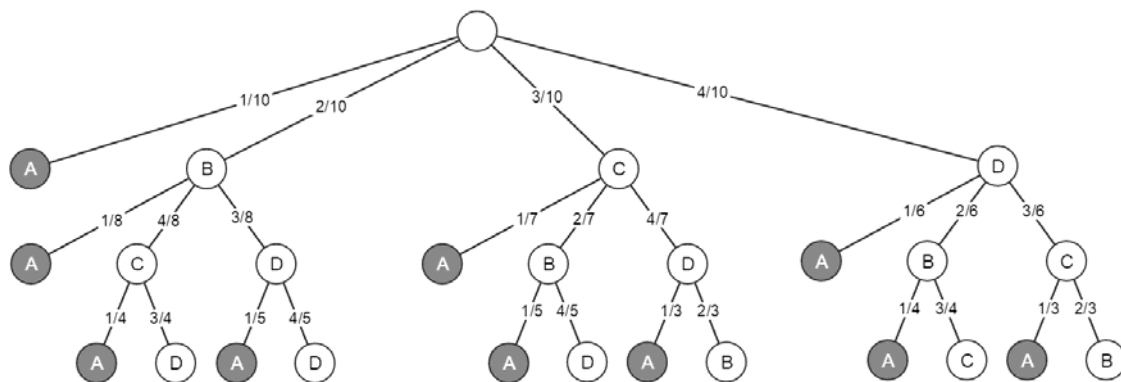
После построения дерева решений необходимо закрасить все узлы, соответствующие искомой записи. В нашей задаче это запись А.

Для расчета вероятности каждого из покрашенных листьев следует перемножить вероятности ребер на пути от листа до корня.

Базируясь на рассчитанных вероятностях покрашенных листьев, можно определить полную вероятность того, что в кэше находится искомая запись. Эта вероятность вычисляется как сумма вероятностей всех покрашенных листьев. Проанализировав частный случай, можно вывести рекуррентную формулу для определения вероятности попадания записи r_i в кэш:

$$P_i = \begin{cases} 0, m = 0, \\ p_i + \sum_{j=1, j \neq i}^n p_j P_i(R \setminus i, CP_{pjn}, m-1, n-1), \end{cases} \quad (15)$$

где CP – функция для вычисления условных вероятностей на основании вектор-строки p и выбранного элемента p_j .



Дерево решений

Функция CP состоит из удаления из вектор-строки указанного элемента с помощью булевой матрицы A и нахождения условной вероятности для оставшихся элементов с помощью матрицы B :

$$CP(p, x, n) = p \times A \times B. \quad (16)$$

Значения для ячеек матрицы A вычисляются по следующей формуле:

$$a_{ij} = \begin{cases} 1, & (i > x \wedge i = j - 1) \vee (i < x \wedge i = j), \\ 0. & \end{cases} \quad (17)$$

Для ячеек матрицы B применяется следующее выражение:

$$b_{ij} = \begin{cases} \frac{1}{1 - p_i}, & i = j, \\ 0. & \end{cases} \quad (18)$$

Используя вероятность попадания в кэш (15), можно вывести формулу для нахождения среднего времени чтения записи r_i :

$$T_i = t_c + t_i(1 - P_i), \quad (19)$$

где t_c – время чтения из кэша; t_i – время чтения записи r_i из базы данных.

Выражение (19) является математической моделью, которая описывает процесс чтения данных из базы данных с использованием системы кэширования. Основываясь на данной модели, можно записать целевую функцию:

$$\sum_{i=1}^n T_i u_i \rightarrow \min. \quad (20)$$

Заключение. В ходе исследования была выведена математическая модель, описывающая процесс чтения из систем кэширования. С помощью нее была получена целевая функция, основывающаяся на минимизации времени чтения записей из базы данных. Минимизация этого времени позволяет приложению получить экономическое и техническое преимущество.

На целевую функцию могут накладываться ограничения для применения методов оптимизации. Описанная математическая модель может служить для создания приложений, которые имеют гарантированный максимальный порог времени чтения из базы данных при высоких нагрузках.

Литература

1. Siewiorek D. Computer Structures: Principles and Examples. NY: Mc Graw Hill, 1981. 960 p.
2. Janert P. Data Analysis with Open Source Tools. Sebastopol: O'Reilly Media, Inc., 2010. 540 p.

References

1. Siewiorek D. Computer Structures: Principles and Examples. NY, Mc Graw Hill, 1981. 960 p.
2. Janert P. Data Analysis with Open Source Tools. Sebastopol, O'Reilly Media, Inc., 2010. 540 p.

Информация об авторе

Цыганенко Никита Павлович – аспирант. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: nikitatsyganenko@belstu.by

Information about the author

Tsyganenko Nikita Pavlovich – PhD student. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: nikitatsyganenko@belstu.by

Поступила 30.11.2016